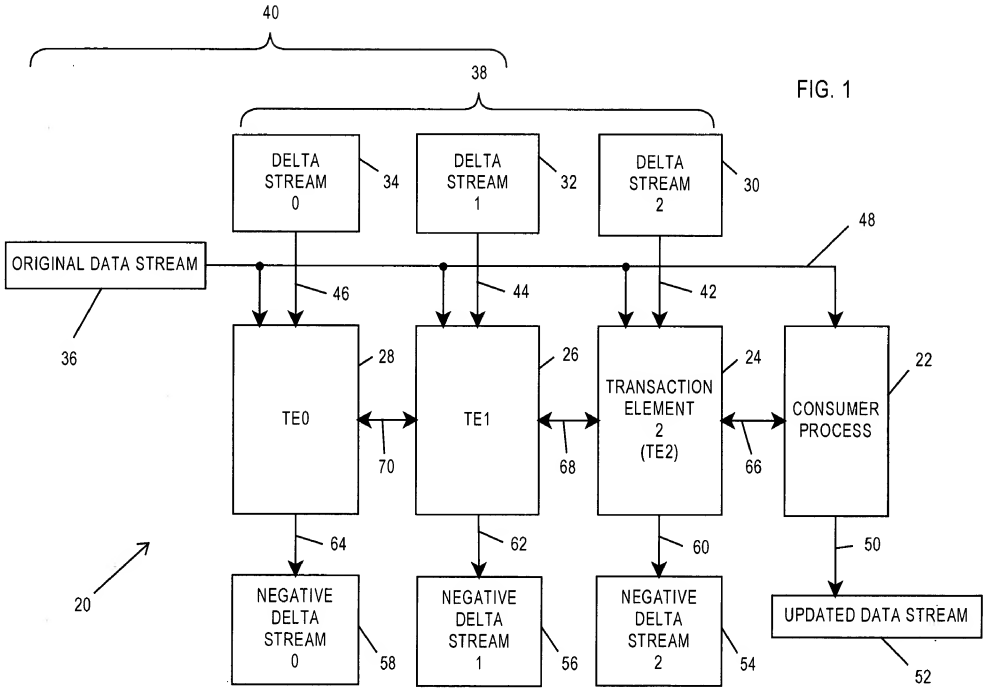


FIG. 1

08/557586

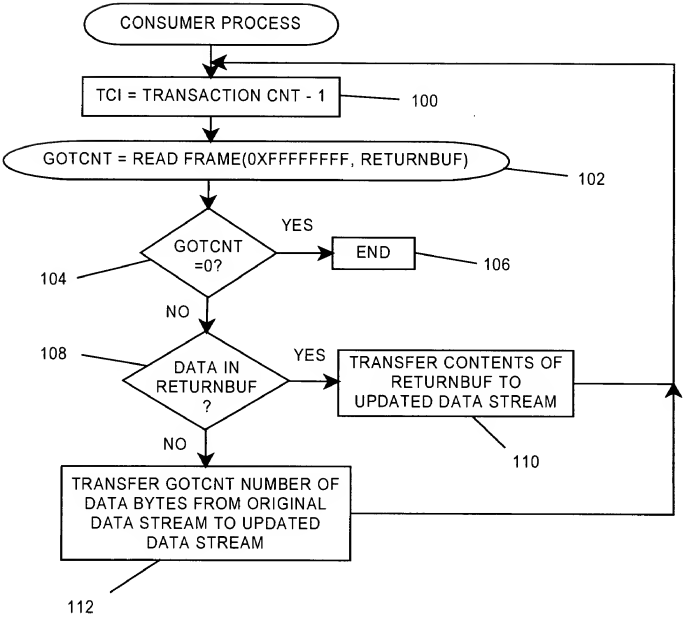
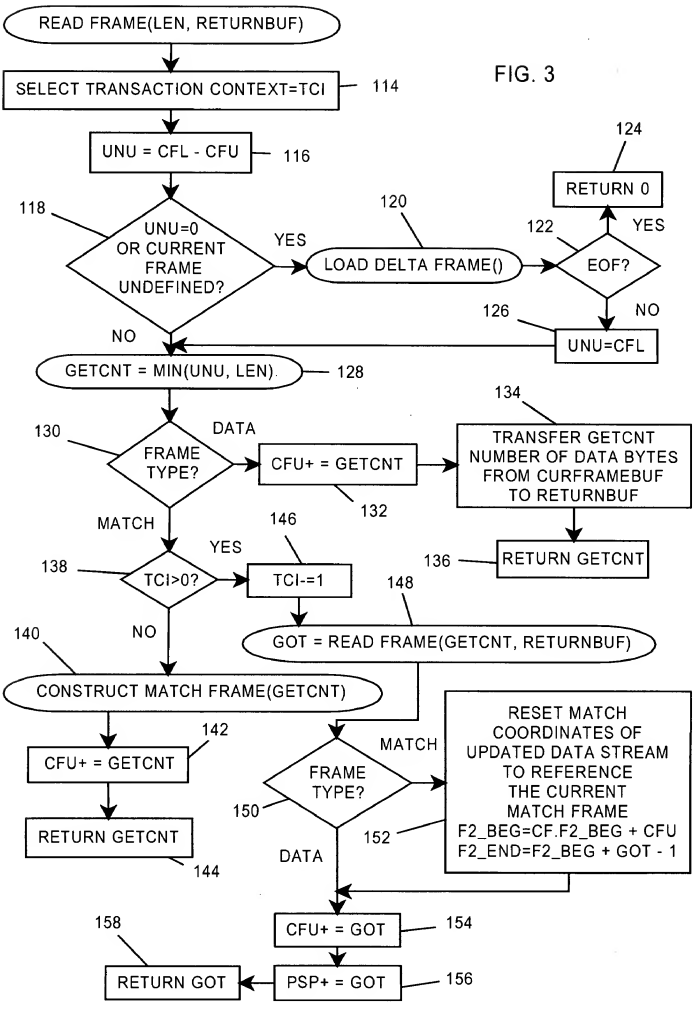

CONSUMER PROCESS
TCI = TRANSACTION CNT - 1
100
GOTCNT = READ FRAME(0XFFFFFFFF, RETURNBUF)
102
GOTCNT =0?
YES
END
106
104
NO
108
DATA IN RETURNBUF ?
YES
TRANSFER CONTENTS OF RETURNBUF TO UPDATED DATA STREAM
110
NO
TRANSFER GOTCNT NUMBER OF DATA BYTES FROM ORIGINAL DATA STREAM TO UPDATED DATA STREAM
112

FIG. 2

FIG. 3


READ FRAME(LEN, RETURNBUF)
SELECT TRANSACTION CONTEXT=TCI
114
UNU = CFL - CFU
116
118
UNU=0 OR CURRENT FRAME UNDEFINED?
YES
120
LOAD DELTA FRAME()
122
EOF?
YES
124
RETURN 0
NO
126
UNU=CFL
NO
GETCNT = MIN(UNU, LEN)
128
130
FRAME TYPE?
DATA
CFU+ = GETCNT
132
134
TRANSFER GETCNT NUMBER OF DATA BYTES FROM CURFRAMEBUF TO RETURNBUF
MATCH
136
RETURN GETCNT
138
TCI>0?
YES
146
TCI-=1
148
GOT = READ FRAME(GETCNT, RETURNBUF)
140
NO
CONSTRUCT MATCH FRAME(GETCNT)
142
CFU+ = GETCNT
RETURN GETCNT
144
FRAME TYPE?
MATCH
150
RESET MATCH COORDINATES OF UPDATED DATA STREAM TO REFERENCE THE CURRENT MATCH FRAME
F2_BEG=CF.F2_BEG + CFU
F2_END=F2_BEG + GOT - 1
152
DATA
158
CFU+ = GOT
154
RETURN GOT
PSP+ = GOT
156

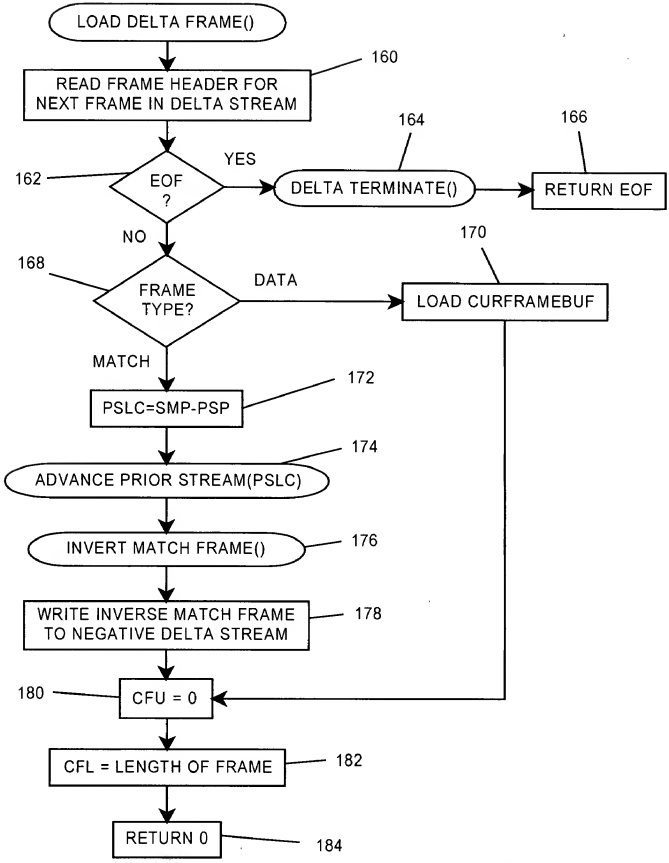

LOAD DELTA FRAME()
READ FRAME HEADER FOR NEXT FRAME IN DELTA STREAM
160
EOF ?
162
YES
164
DELTA TERMINATE()
166
RETURN EOF
NO
FRAME TYPE?
168
DATA
170
LOAD CURFRAMEBUF
MATCH
PSLC=SMP-PSP
172
ADVANCE PRIOR STREAM(PSLC)
174
INVERT MATCH FRAME()
176
WRITE INVERSE MATCH FRAME TO NEGATIVE DELTA STREAM
178
CFU = 0
180
CFL = LENGTH OF FRAME
182
RETURN 0
184

FIG. 4

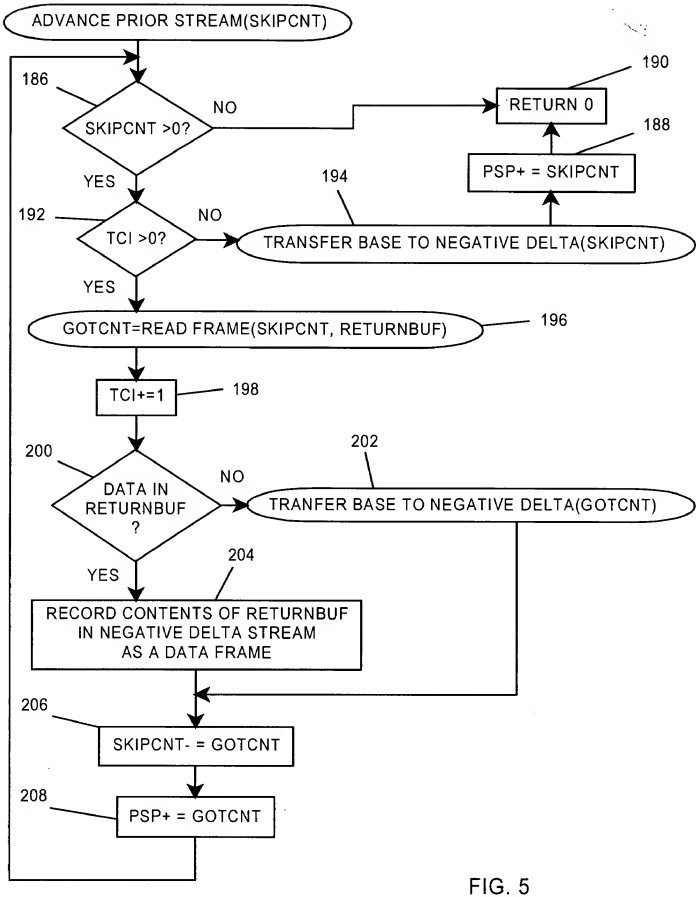

ADVANCE PRIOR STREAM(SKIPCNT)
186
SKIPCNT >0?
NO
190
RETURN 0
188
PSP+ = SKIPCNT
YES
194
192
TCI >0?
NO
TRANSFER BASE TO NEGATIVE DELTA(SKIPCNT)
YES
GOTCNT=READ FRAME(SKIPCNT, RETURNBUF)
196
TCI+=1
198
200
DATA IN RETURNBUF ?
NO
202
TRANFER BASE TO NEGATIVE DELTA(GOTCNT)
YES
204
RECORD CONTENTS OF RETURNBUF IN NEGATIVE DELTA STREAM AS A DATA FRAME
206
SKIPCNT- = GOTCNT
208
PSP+ = GOTCNT

FIG. 5

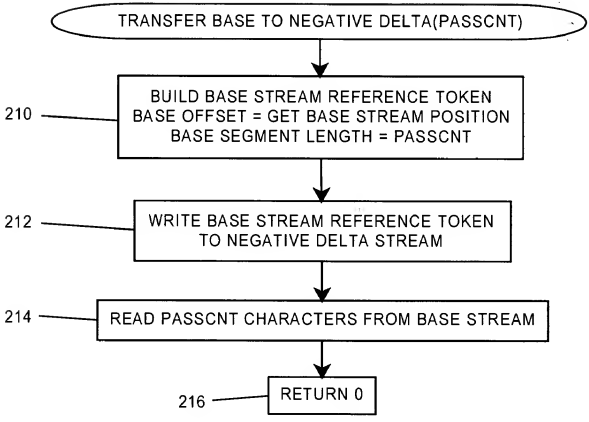

TRANSFER BASE TO NEGATIVE DELTA(PASSCNT)
210
BUILD BASE STREAM REFERENCE TOKEN
BASE OFFSET = GET BASE STREAM POSITION
BASE SEGMENT LENGTH = PASSCNT
212
WRITE BASE STREAM REFERENCE TOKEN
TO NEGATIVE DELTA STREAM
214
READ PASSCNT CHARACTERS FROM BASE STREAM
216
RETURN 0

FIG. 6

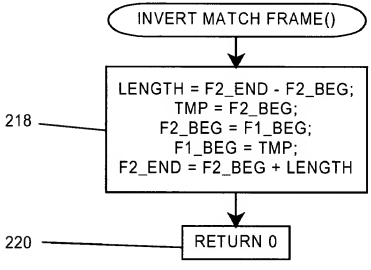

INVERT MATCH FRAME()
218
LENGTH = F2_END - F2_BEG;
TMP = F2_BEG;
F2_BEG = F1_BEG;
F1_BEG = TMP;
F2_END = F2_BEG + LENGTH
220
RETURN 0

FIG. 7

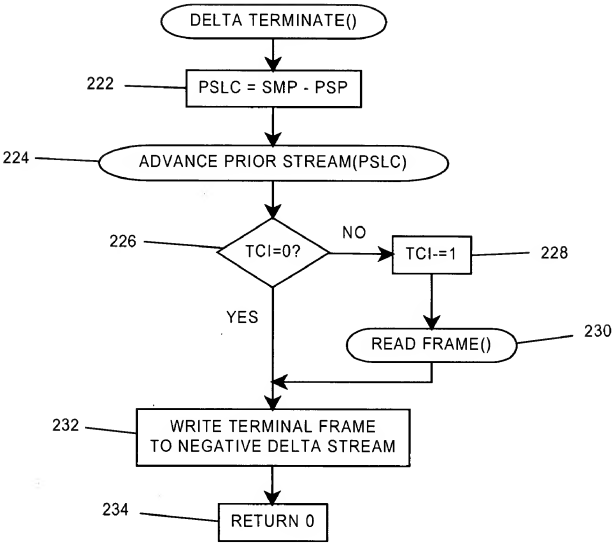

DELTA TERMINATE()
222
PSLC = SMP - PSP
224
ADVANCE PRIOR STREAM(PSLC)
226
TCI=0?
NO
TCI-=1
228
YES
READ FRAME()
230
232
WRITE TERMINAL FRAME
TO NEGATIVE DELTA STREAM
234
RETURN 0

FIG. 8

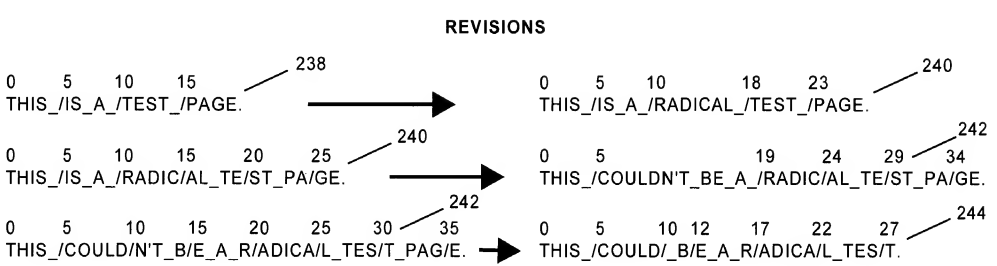
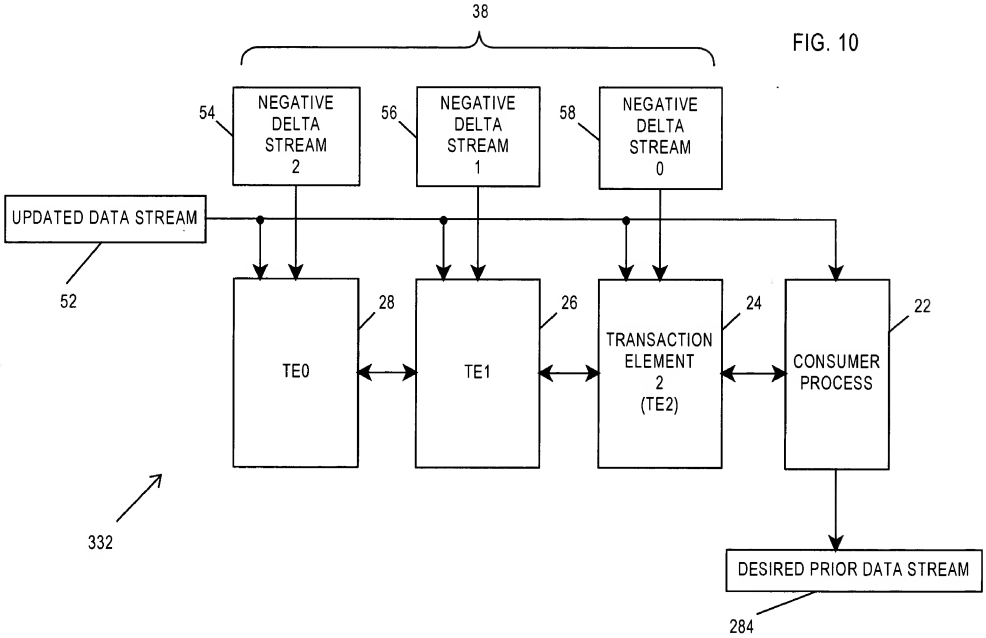


FIG. 9

08/557586

FIG. 10



08/557586

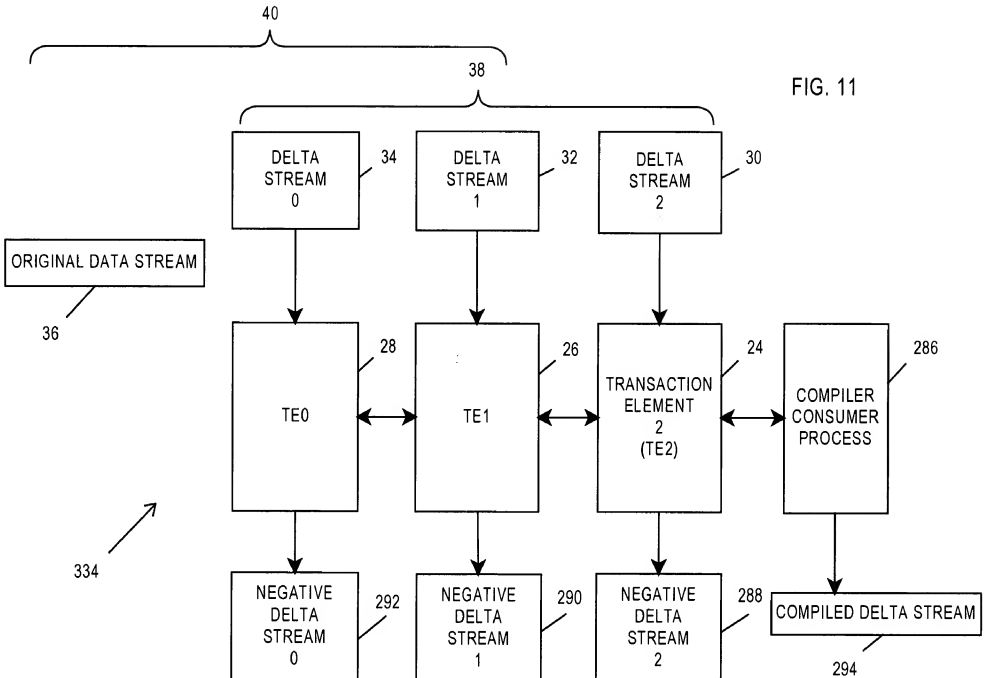


FIG. 11

08/557586

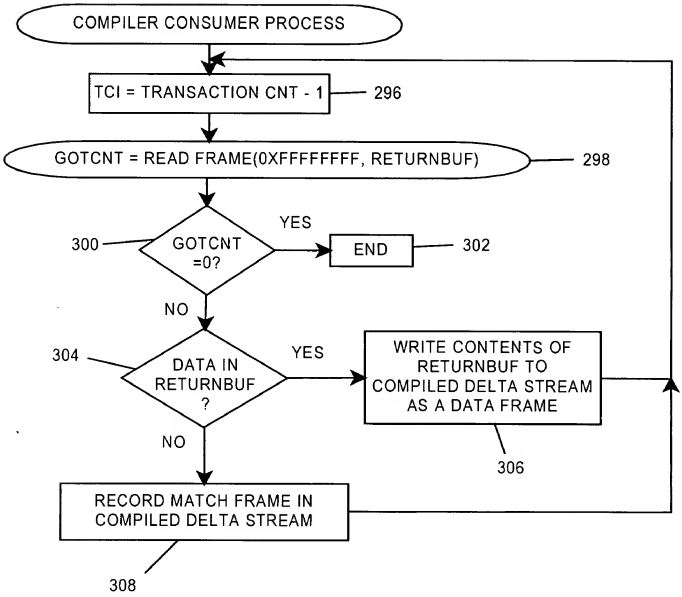

COMPILER CONSUMER PROCESS
TCI = TRANSACTION CNT - 1
296
GOTCNT = READ FRAME(0XFFFFFFFF, RETURNBUF)
298
300
GOTCNT =0?
YES
END
302
NO
304
DATA IN RETURNBUF ?
YES
WRITE CONTENTS OF RETURNBUF TO COMPILED DELTA STREAM AS A DATA FRAME
306
NO
RECORD MATCH FRAME IN COMPILED DELTA STREAM
308

FIG. 12

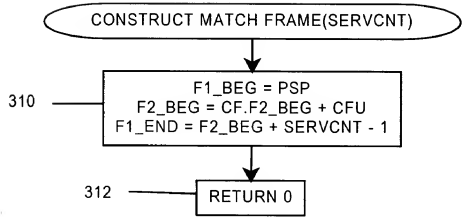

CONSTRUCT MATCH FRAME(SERVCNT)
310
F1_BEG = PSP
F2_BEG = CF.F2_BEG + CFU
F1_END = F2_BEG + SERVCNT - 1
312
RETURN 0

FIG. 13

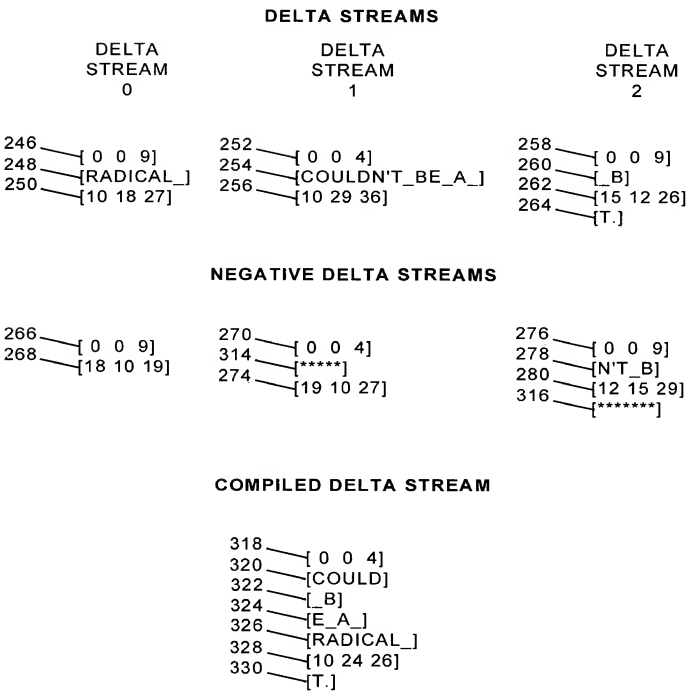

DELTA STREAMS
DELTA STREAM 0
246 [0 0 9]
248 [RADICAL_]
250 [10 18 27]
DELTA STREAM 1
252 [0 0 4]
254 [COULDN'T_BE_A_]
256 [10 29 36]
DELTA STREAM 2
258 [0 0 9]
260 [_B]
262 [15 12 26]
264 [T.]
NEGATIVE DELTA STREAMS
266 [0 0 9]
268 [18 10 19]
270 [0 0 4]
314 [*****]
274 [19 10 27]
276 [0 0 9]
278 [N'T_B]
280 [12 15 29]
316 [*******]
COMPILED DELTA STREAM
318 [0 0 4]
320 [COULD]
322 [_B]
324 [E_A_]
326 [RADICAL_]
328 [10 24 26]
330 [T.]

FIG. 14